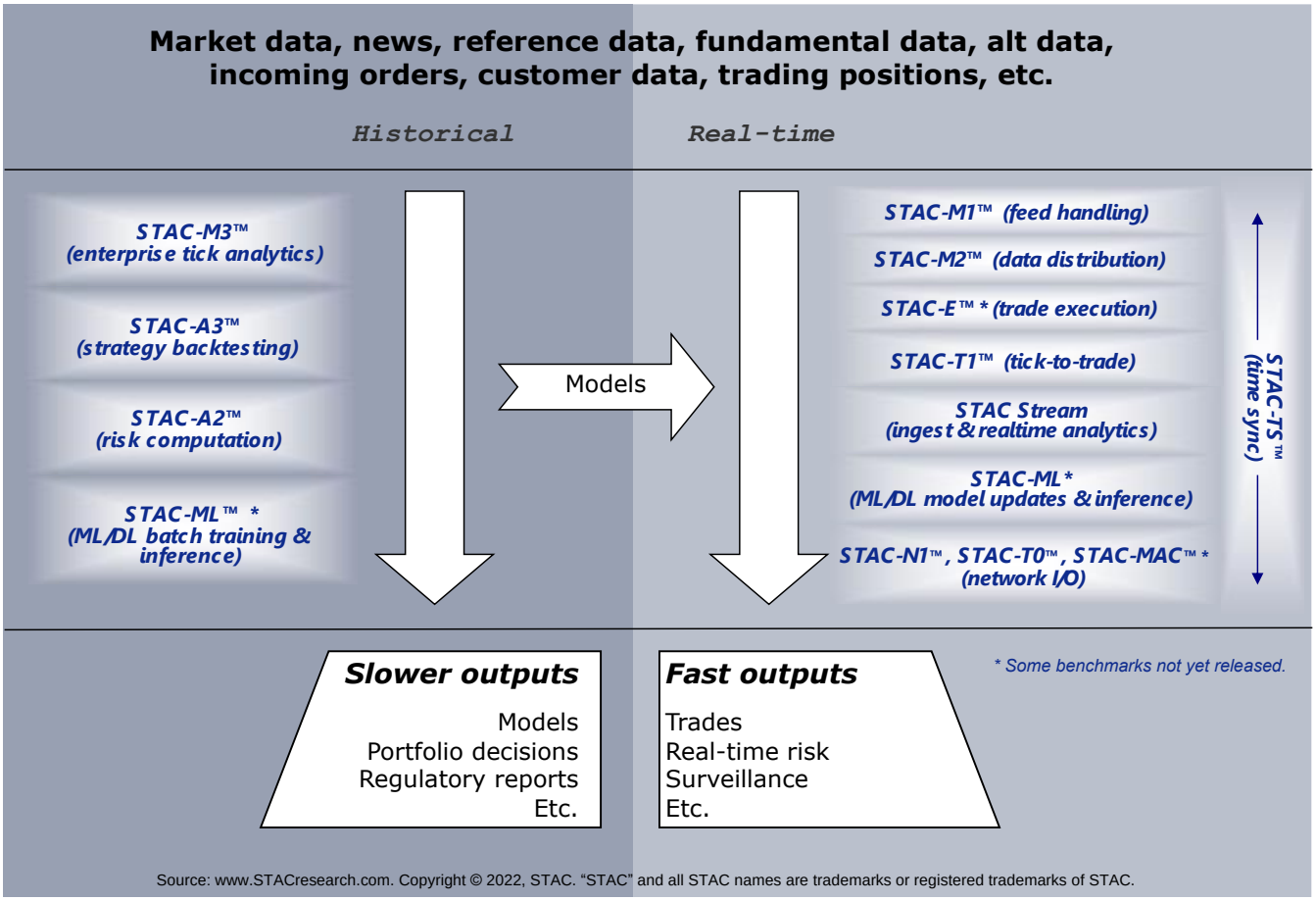
Capital markets workload categorization

As the figure below shows, STAC Benchmarks™ for capital markets cover key workloads in which financial institutions turn historical and realtime information into action and knowledge. These workloads vary in their intensity of computation, storage I/O, and network I/O. The benchmark suites are categorized primarily as market data (STAC-M™), analytics (STAC-A™), machine learning (STAC-ML™), execution (STAC-E™), "tick-to-trade" (STAC-T™), and time sync (STAC-TS™) with domains that are roughly based on the way that user firms think about their architectures—that is, how they group functional requirements today, divide their technical organizations, and buy products.

STAC® domains in capital markets



The following tables summarize each benchmark domain. Domains are grouped by whether they focus on a "Fast Workload" (relatively simple computation, with speed of communication being paramount) or a "Big Workload" (where the size of the dataset that must be moved or analyzed, or the magnitude of computation is the biggest challenge—i.e., "big data" or "big compute"). For details or to read STAC Reports™ from the domain, click its title to visit its page on the STAC® site.

Big Workloads

Domain	Caption	Workload summary	Example metrics	Example products that can be tested
STAC-M3	Tick data analytics	Applying analytics to market data time series. Use cases range from execution reporting and risk management to strategy development. Historical and real-time processing and analytics.	• Query-response time • Application-level I/O throughput • Storage efficiency • Data availability	• Tick databases • Storage systems • File systems • Cloud platforms
STAC-A3	Strategy backtesting	Testing parameterized trading algorithms against historical data.	• Speed: time to compute a fixed workload • Capacity: max workload computed in a fixed time • Quality: adherence of results to theoretical values • Efficiency: output per unit of power/space/money	• Storage systems • File systems • Database software • Cloud platforms
STAC-A2	Risk computations	Computing risk metrics for options, viewed as an end-to-end process and also broken down into discrete compute steps.	• Speed: time to compute a fixed workload • Capacity: max workload computed in a fixed time • Quality: adherence of results to theoretical values • Efficiency: output per unit of power/space/money	• CPUs, GPUs, FPGAs • Programming tools • Analytic libraries • Grid software • Clouds
STAC-ML (batch training and inference subset)	Artificial intelligence (*some not yet released*)	Batch-mode inference tasks, as well as offline training of machine learning and deep learning workloads.	• Speed: latency of inference • Quality: “goodness” of results • Time to market: cycle time for model training • Efficiency: price performance in the public cloud or output per unit of on-prem resource	• ML/DL frameworks • CPUs, GPUs, FPGAs, custom processors • Inference engines • Libraries • Scaling frameworks • Cloud: IaaS, MLaaS • Storage infrastructure • Techniques

Fast Workloads

Domain	Caption	Workload summary	Example metrics	Example products that can be tested
STAC-T0	Network I/O for trading (black-box approach)	Transmitting messages via sockets. Messages follow a tick-to-trade pattern.	• "Actionable" latency • FIFO/LIFO/FILO latency	• FPGA platforms and IP cores • Software solutions • NICs/servers/switches
STAC-T1	Tick to trade	Wire-to-wire I/O performance of a trading system including market data parsing and order generation and transmission.	• Market data to order latency at various multiples of market rates	• Feed handling and market gateway software • Server/OS/networking
STAC-N1	Network I/O for trading (software-based)	Transmitting messages directly via a network API (sockets, RDMA, other). Messages follow a market data pattern.	• Per-message API-to-API latency (half round trip) • Max throughput • CPU and memory utilization of sender and receiver	• Network cards and drivers (e.g., kernel bypass) • Server/OS/switches
"STAC MAC"	MAC/PHY logic (*not yet released*)	Handling just the Ethernet MAC & PHY functions	• Latency	• IP cores for FPGA
Under considera-tion	Network mux (*not yet released*)	Multiplexing TCP messages such as orders	• Latency	• Multiplexing switches
STAC-TS	Time synchronization	Synchronizing time, timestamping events, capturing events. Reporting traceability of timestamping points in an enterprise to a standard (e.g., UTC, NIST).	• Accuracy of application timestamps relative to an external reference • Accuracy of application timestamps relative to the host clock • Accuracy of master clocks under exception conditions • Accuracy of network timestamps • Capacity of network capture	• Timestamping API calls • Master clocks • PTP solutions • NTP solutions • Switches • NICs • Capture cards • Aggregation devices
STAC-M1	Direct feed integration	Taking inbound market data messages from exchanges, normalizing and caching them, and making them available via an API.	• Latency from exchange message hitting the wire to normalized update coming through API • Max throughput, and latency at that throughput • CPU/memory consumption of client	• Ticker plant software • Ticker plant appliances • Acceleration cards • Server/OS/networking
STAC-M2	Market data distribution	Taking normalized market data streams through a publisher API and delivering them to multiple consumers via a subscriber API.	• Latency from the moment a message is ready for distribution to the moment it exits the subscriber API. • Latency to "undisturbed consumers" when other consumers are starved of resources	• Messaging software • Messaging appliances • Market data platforms • Server/OS/networking

STAC-E	Trade-execution (*not yet released*)	Performing functions on trade messages inbound via APIs and/or wire protocols and transmitting them to a destination. Functions may include validation, pre-trade risk and compliance checks, order-state management, matching, and smart order routing. Includes internal messaging via persistent methods (resilience to internal failures).	• One-way latency according to message type • Two way latency (e.g., order to fill, cancel to ACK, order to quote) • Internal API-to-API latency of persistent messaging • Max throughput	• Market- or client-facing execution gateways • Messaging software, messaging appliances • OMS • Smart order routers • Matching engines • Server/OS/networking
STAC-M3 "Jalua"	Streaming ingest and analytics	Various use cases that require ingest of streaming data and either event-driven or on-demand analytics.	• Capacity: Max ingest rate • Latency: time for inbound data to become available for analytics or to result in outbound notifications • Efficiency: work per unit of power/space/dollar	• Data management software • Memory, storage, interconnects • Deployed systems and clouds
STAC-ML (model updates and inference subset)	Artificial intelligence (*not yet released*)	Batch-mode inference tasks, as well as offline training of machine learning and deep learning workloads.	• Speed: latency of inference • Quality: "goodness" of results • Time to market: cycle time for model training • Efficiency: price performance in the public cloud or output per unit of on-prem resource	• ML/DL frameworks • CPUs, GPUs, FPGAs, custom processors • Inference engines • Libraries • Scaling frameworks • Cloud: IaaS, MLaaS • Storage infrastructure • Techniques